The Road to Joy is DAMmed!

Digital Collections Inundated

- AI has joined the deluge of bots/crawlers
- No longer effective methods:
  - IP blocking
  - Rate-limiting
  - Flitering by user agents
- Login restrictions not always an option

What can we do???

DAM the flood!

- **Proof-of-work challenges**
  - **Server picks random number**
  - **Creates a unique key**
  - **Joins the two and changes them to create a “solution”**
  - **Key and solution are sent to the client**
  - **The client has to find the random number through trial and error**

How does this protect against traffic?

- Many bots don’t load or run javascript
- Won’t see or try to solve the challenge
- Those that do will spend resources on these challenges

DAM = Django-Altcha-Middleware

- Lovingly referred to as “daaaaam!”
- Can be used in any Django project
- Protects all URLs (pages) or specific ones
- Uses Altcha to generate challenges, implement the widget, and check solutions
- Can be used as a blueprint for similar apps

The way it works

- It denies access to the protected pages until the client solves the challenge
- Exceptions are configurable
- Non-interactive and quick
- Correct solutions get content
- Won’t be challenged again for a while

Results

- A lifeline for our drowning devs
- 97% - 83% avg. traffic blocked
- Easy to see what’s happening in logs
- Reduced frequency of service-impacting events
- Easy and quick to adapt

DAM the flood!

https://github.com/unt-libraries/django-altcha-middleware